Amendments to the Claims

The listing of claims will replace all prior versions and listings of claims in the application.

1. (previously presented) A method of generating a Service Ticket for a requested Service comprising:

receiving a request for a Service Ticket from a client;

generating a session key;

encrypting a cipher text with the session key

determining a number of servers designated to provide the requested service;

for each providing server, encrypting the session key with a secret key associated with each respective server;

creating a Service Ticket that includes an encrypted session key for each providing server, and the encrypted cipher text; and

transmitting the Service Ticket to the client.

2. (previously presented) The method of claim 1, further including:

generating a Ticket-Granting-Ticketing utilizing a protocol substantially in compliance with the Kerberos protocol; and

wherein receiving a request for a Service Ticket from a client further includes receiving the Ticket-Granting-Ticket from the client.

3. (previously presented) The method of claim 1, wherein determining the number of servers designated to provide the requested service includes:

utilizing a database that maps a generic server name to a specific server name; and

setting the numbers of servers designated to provide the service equal to the number of specific server names mapped to the generic server name that provides the requested service.

4. (previously presented) The method of claim 3, wherein utilizing a database that maps a generic server name to a specific server name includes selecting a database from a group consisting essentially of:

a domain name server database,

a database associated with a Key Distribution Center, and

a Kerberos database.

5. (previously presented) The method of claim 3, wherein the secret keys associated with each providing server are not synchronized across the providing servers.

6. (previously presented) The method of claim 1, wherein the created Service Ticket includes:

a header that designates the Service Ticket as a format that includes multiple encrypted session keys,

a field that expressly designates the number of encrypted session keys,

an encrypted session key for each providing server, and

the encrypted cipher text.

7. (previously presented) The method of claim 1, further including:

determining if the requested service is provided by a plurality of servers:

if not, generating the Service Ticket utilizing a single server mode; and

if so, generating the Service Ticket as described in claim 1.

8. (previously presented) The method of claim 7, wherein generating the Service Ticket utilizing a single server mode includes:

generating a cipher text;

encrypting the cipher text with a secret key associated with the providing server; and

transmitting the Service Ticket, that includes the encrypted cipher text, to the client.

9. (previously presented) A method of authenticating a client's request for a service provided by a service pool comprising;

a server receiving a Service Ticket having at least one encrypted session key, and an encrypted cipher text;

decrypting the encrypted session key associated with the receiving server utilizing a secret key associated with the receiving server;

decrypting the cipher text utilizing the decrypted session key; and

providing the service to the client.

10. (previously presented) The method of claim 9, wherein receiving a Service Ticket is part of a series of client transactions substantially in compliance with the Kerberos protocol.

11. (previously presented) The method of claim 9, wherein decrypting the encrypted session key includes:

determining the number of encrypted session keys included within the received Service Ticket;

for each encrypted session key, decrypting the encrypted session key utilizing a secret key associated with the receiving server; and

wherein decrypting the cipher text utilizing the decrypted session key includes

for each encrypted session key, attempting to decrypt the cipher text with the decrypted session key;

if the cipher text is successfully decrypted, providing the service to the client.

12. (previously presented) The method of claim 9, wherein decrypting the encrypted session key associated with the receiving server utilizing a secret key associated with the receiving server includes:

utilizing a server identifier to determine which encrypted session key is associated with the receiving server; and

decrypting the associated encrypted session key utilizing a secret key associated with the receiving server.

13. (previously presented) The method of claim 9, further including:

determining if the received Service Ticket includes a plurality of encrypted

session keys for multiple servers

if not, processing the ticket in a single server mode; and

if so, processing the ticket as described in claim 9.

14. (previously presented) The method of claim 13, wherein processing the ticket in a single server mode includes processing the Service Ticket in utilizing a process substantially compliant with the Kerberos protocol.

15. (previously presented) The method of claim 9, wherein receiving a Service Ticket includes:

a managing agent receiving a Service Ticket;

the managing agent selecting a receiving server from a server pool having a plurality of servers;

routing the Service Ticket to the receiving server.

16. (previously presented) The method of claim 15, wherein the plurality of servers each include a secret key associated with the respective servers, and the plurality of secret keys are not synchronized among the plurality of servers..

17. (previously presented) The method of claim 16, wherein the server pool functions as a group of independent computers working together as a single system.

Claims 18-33 (cancelled)

34. (previously presented) An article comprising:

a storage medium having a plurality of machine accessible instructions, wherein when the instructions are executed, the instructions provide for:

receiving a request for a Service Ticket from a client;

generating a session key;

encrypting a cipher text with the session key

determining the number of servers designated to provide the requested service;

for each providing server, encrypting the session key with a secret key associated with each respective server;

creating a Service Ticket that includes an encrypted session key for each providing server, and the encrypted cipher text; and

transmitting the Service Ticket to the client.

35. (previously presented) The article of claim 34, further including instructions providing for:

generating a Ticket-Granting-Ticketing utilizing a protocol substantially in compliance with the Kerberos protocol; and

wherein receiving a request for a Service Ticket from a client further includes receiving the Ticket-Granting-Ticket from the client.

36. (previously presented) The article of claim 34, wherein the instructions providing for determining the number of servers designated to provide the

requested service includes instructions providing for:

utilizing a database that maps a generic server name to a specific server name; and

setting the numbers of servers designated to provide the service equal to the number of specific server names mapped to the generic server name that provides the requested service.

37. (previously presented) The article of claim 36, wherein the instructions providing for utilizing a database that maps a generic server name to a specific server name includes instructions providing for selecting a database from a group consisting essentially of:

a domain name server database,

a database associated with a Key Distribution Center, and

a Kerberos database.

38. (previously presented) The article of claim 36, wherein the secret keys associated with each providing server are not synchronized across the providing servers.

39. (previously presented) The article of claim 38, wherein the instructions providing for creating a Service Ticket further includes instructions providing for creating a Service Ticket that includes:

a header that designates the Service Ticket as a format that includes multiple encrypted session keys,

a field that expressly designates the number of encrypted session keys,

an encrypted session key for each providing server, and

the encrypted cipher text.

40. (previously presented) The article of claim 34, further including instructions providing for:

determining if the requested service is provided by a plurality of servers:

if not, generating the Service Ticket utilizing a single server mode; and

if so, generating the Service Ticket as described in claim 1.

41. (previously presented) The article of claim 40, wherein the instructions providing for generating the Service Ticket utilizing a single server mode includes instructions providing for:

generating a cipher text;

encrypting the cipher text with a secret key associated with the providing server; and

transmitting the Service Ticket, that includes the encrypted cipher text, to the client.

42. (previously presented) An article comprising:

a storage medium having a plurality of machine accessible instructions, wherein when the instructions are executed, the instructions provide for:

a server receiving a Service Ticket having at least one encrypted session key, and an encrypted cipher text;

decrypting the encrypted session key associated with the receiving server utilizing a secret key associated with the receiving server;

decrypting the cipher text utilizing the decrypted session key; and

providing the service to the client.

43. (previously presented) The article of claim 42, wherein the instructions provide for receiving a Service Ticket are part of a series of client transactions substantially in compliance with the Kerberos protocol.

44. (previously presented) The article of claim 42, wherein the instructions provide for decrypting the encrypted session key includes instructions provide for:

determining the number of encrypted session keys included within the received Service Ticket;

for each encrypted session key, decrypting the encrypted session key utilizing a secret key associated with the receiving server; and

wherein decrypting the cipher text utilizing the decrypted session key includes

for each encrypted session key, attempting to decrypt the cipher text with the decrypted session key;

if the cipher text is successfully decrypted, providing the service to the client.

45. (previously presented) The article of claim 42, wherein the instructions provide for decrypting the encrypted session key associated with the receiving server utilizing a secret key associated with the receiving server includes instructions provide for:

utilizing a server identifier to determine which encrypted session key is

associated with the receiving server; and

decrypting the associated encrypted session key utilizing a secret key associated with the receiving server.

46. (previously presented) The article of claim 42, further including instructions provide for:

determining if the received Service Ticket includes a plurality of encrypted session keys for multiple servers

if not, processing the ticket in a single server mode; and

if so, processing the ticket as described in claim 9.

47. (previously presented) The article of claim 46, wherein the instructions provide for processing the ticket in a single server mode includes instructions provide for processing the Service Ticket in utilizing a process substantially compliant with the Kerberos protocol.

48. (previously presented) The article of claim 42, wherein the instructions provide for receiving a Service Ticket includes instructions provide for:

a managing agent receiving a Service Ticket;

the managing agent selecting a receiving server from a server pool having a plurality of servers;

routing the Service Ticket to the receiving server.

49. (previously presented) The article of claim 48, wherein the plurality of

servers each include a secret key associated with the respective servers, and the plurality of secret keys are not synchronized among the plurality of servers..

50. (previously presented) The article of claim 49, wherein the server pool functions as a group of independent computers working together as a single system.